



Knowledge for Tomorrow

Visualizing Provenance using Comics

Andreas Schreiber

German Aerospace Center (DLR)

Regina Struminski

University of Applied Science Düsseldorf





Introduction



**Deutsches Zentrum
für Luft- und Raumfahrt**

German Aerospace Center

Simulation and Software Technology, Cologne/Berlin

Head of Intelligent and Distributed Systems department

Institute of Data Science, Jena

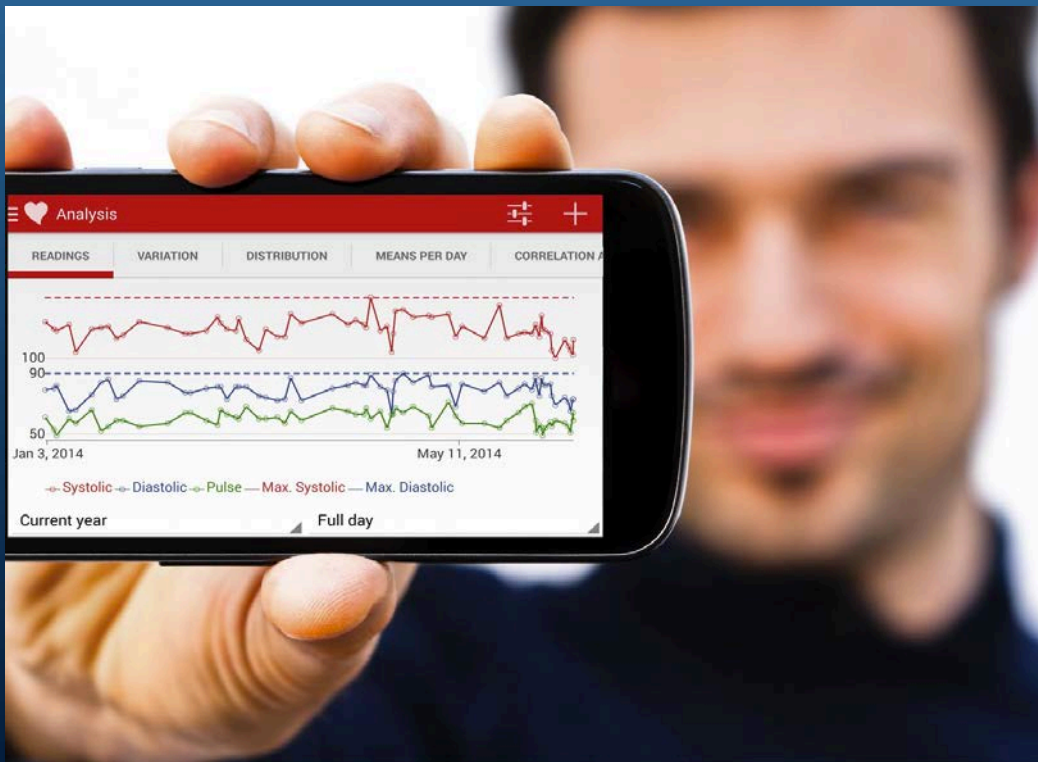
Head of Secure Software Engineering group



**Co-Founder
Data Scientist
Patient**

Motivation – Use Cases

Quantified Self (n = 1 participant)

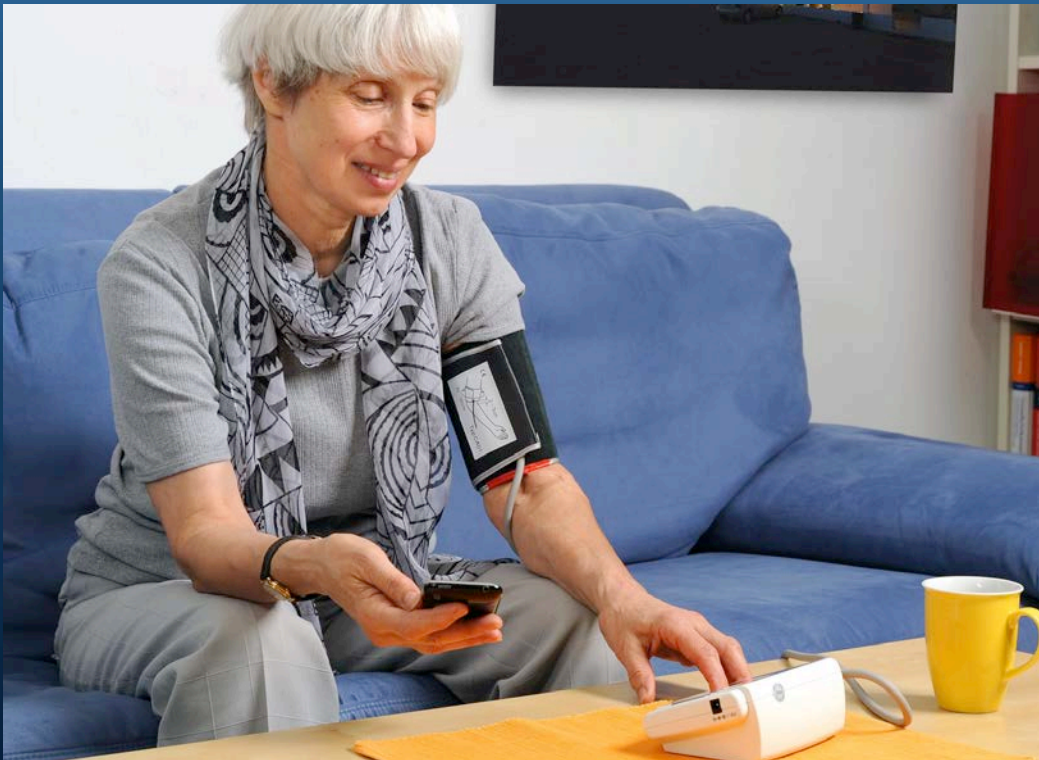


Medical Trials (n > 1 participants)



Motivation – Use Cases

Telemedicine

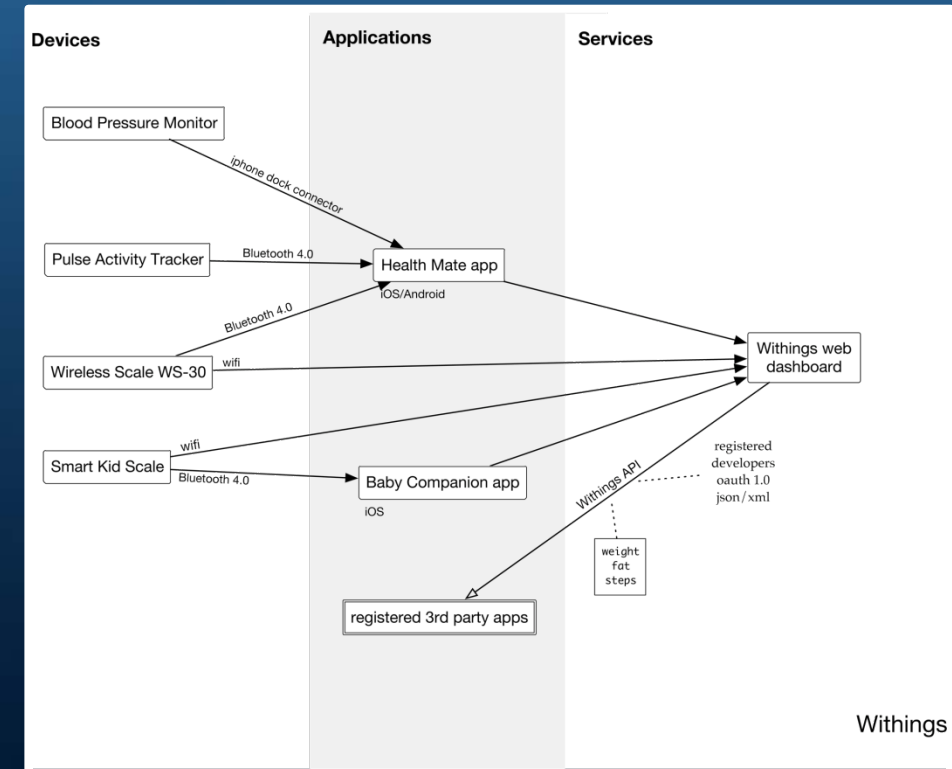
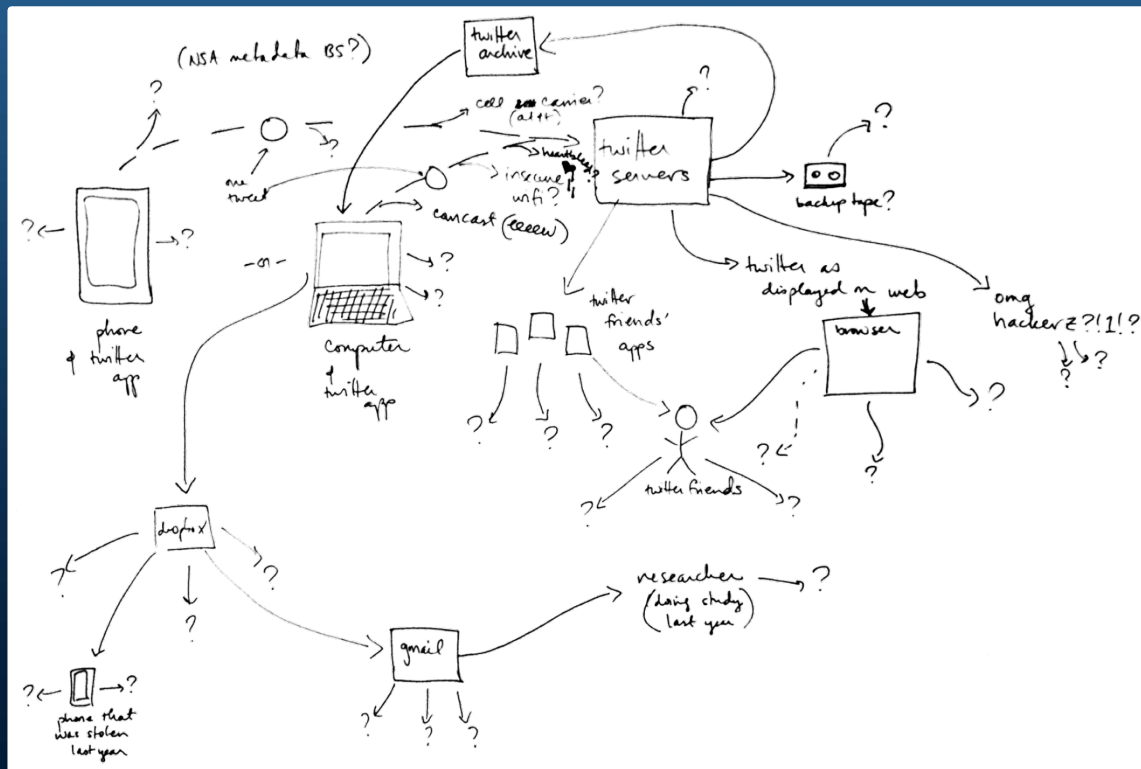


Medical experiments

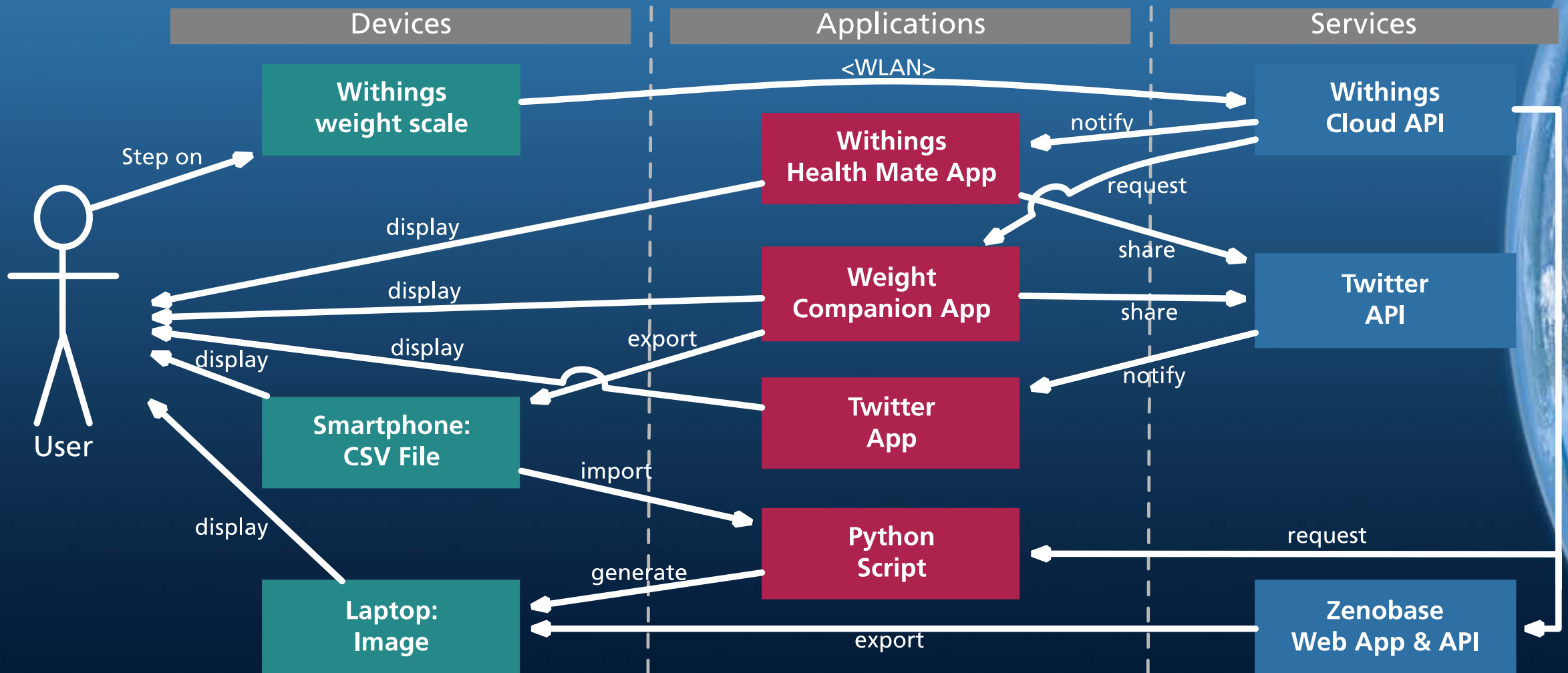


Understand, how Quantified Self data has been produced, processed, stored, accessed, ...

Pictures from *Breakout Session on Mapping Data Access* (2014 QS Europe Conference, Amsterdam)
<https://forum.quantifiedself.com/t/breakout-mapping-data-access/995>



Example: Weight Tracking Workflow



Questions related to Quantified Self Data and Activities

Data

- What data about the user were created during the activity X?
- What data about the user were automatically generated?
- What data about the user were derived from manual input?

Apps and Services

- Which activities support visualization of the users data?
- In which activities can the user input data?
- What processes are communicating data?

Access and Privacy

- What parties were involved in generating data X?
- What parties got access on data X?
- Can other parties see user's data X?

Provenance Model for Quantified Self

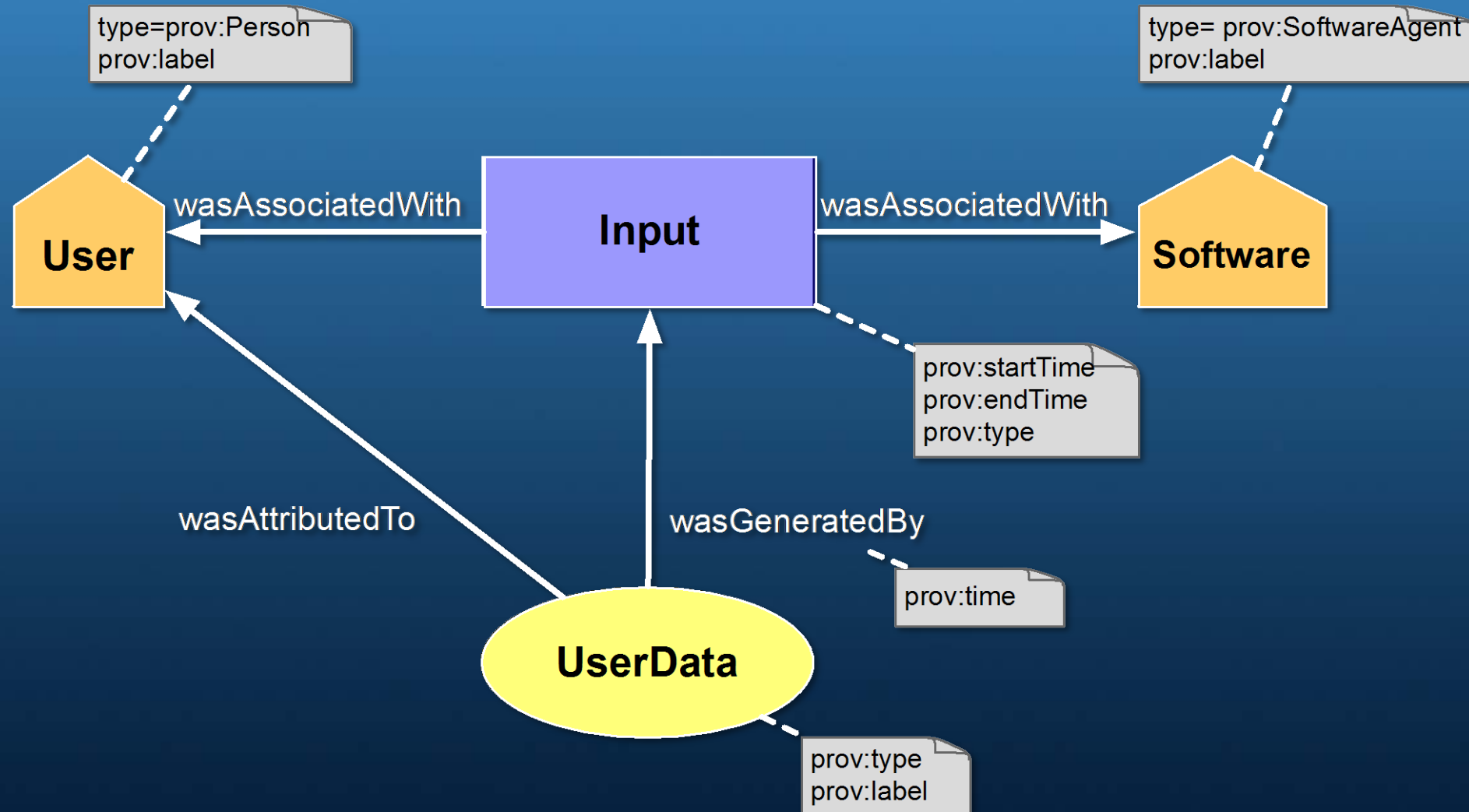
Sub models for basic *Activities*

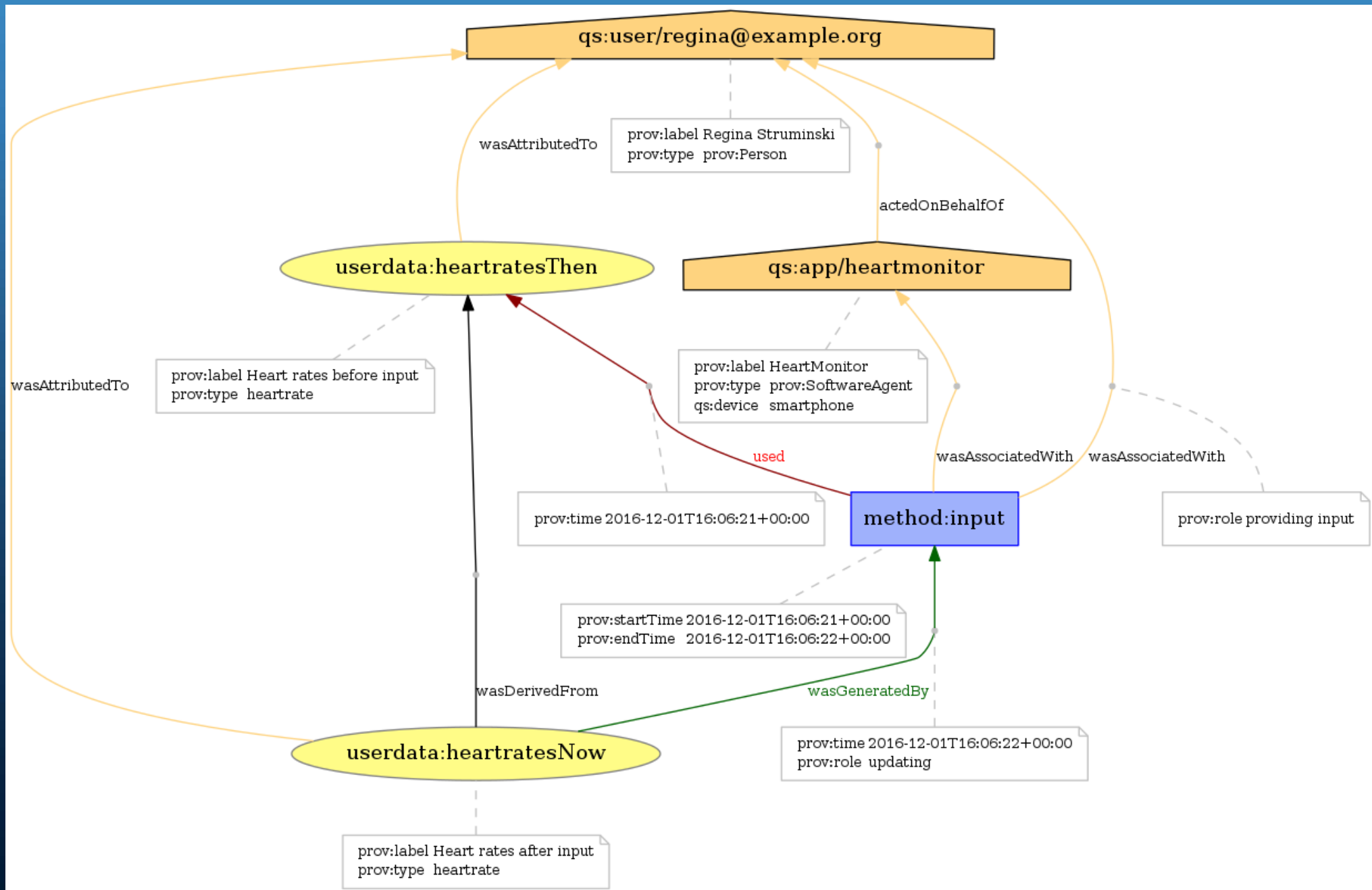
- Input
- Sensing
- Export
- Request
- Aggregate
- Visualize

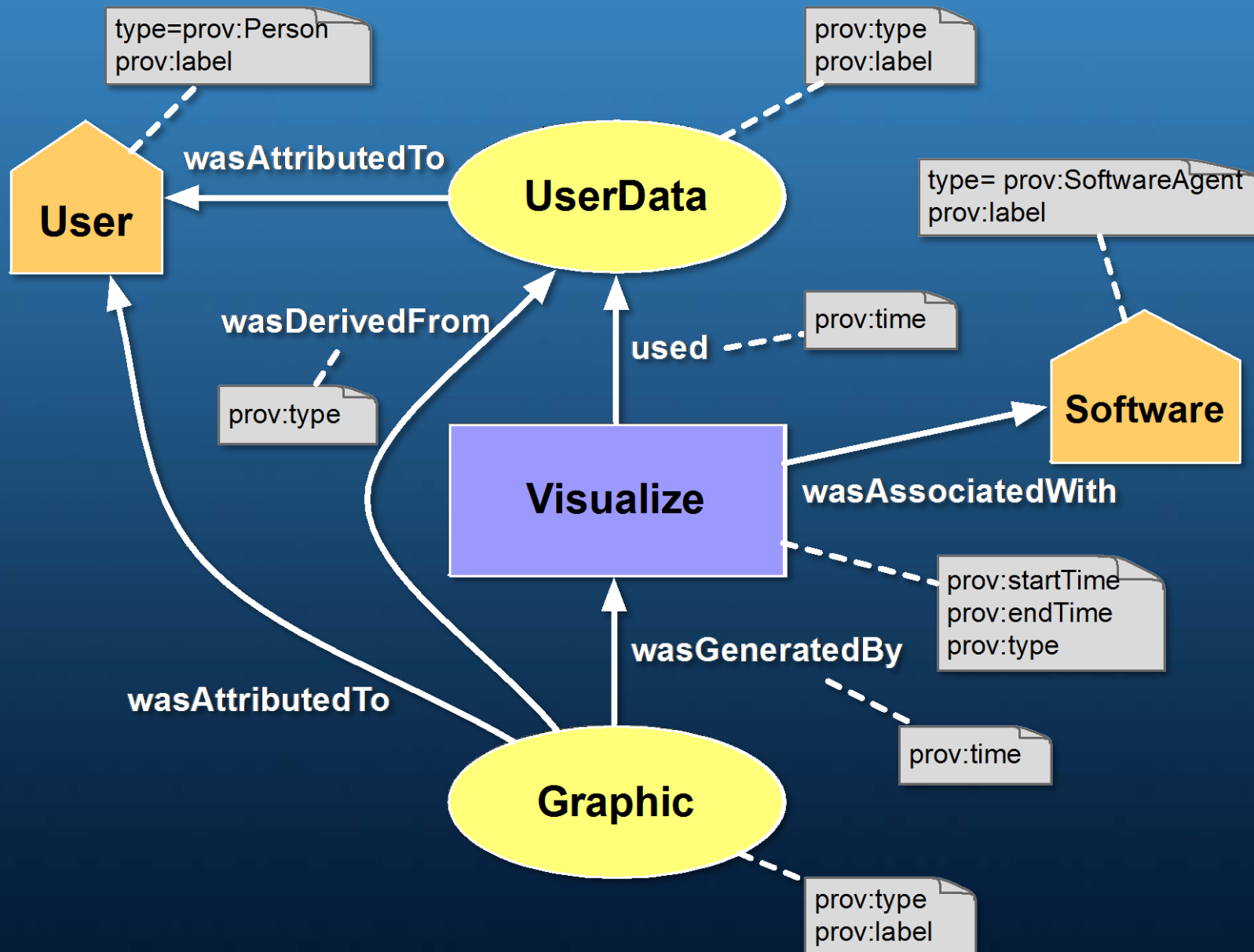
The activities generate or change data that is associated or attributed to *Agents*

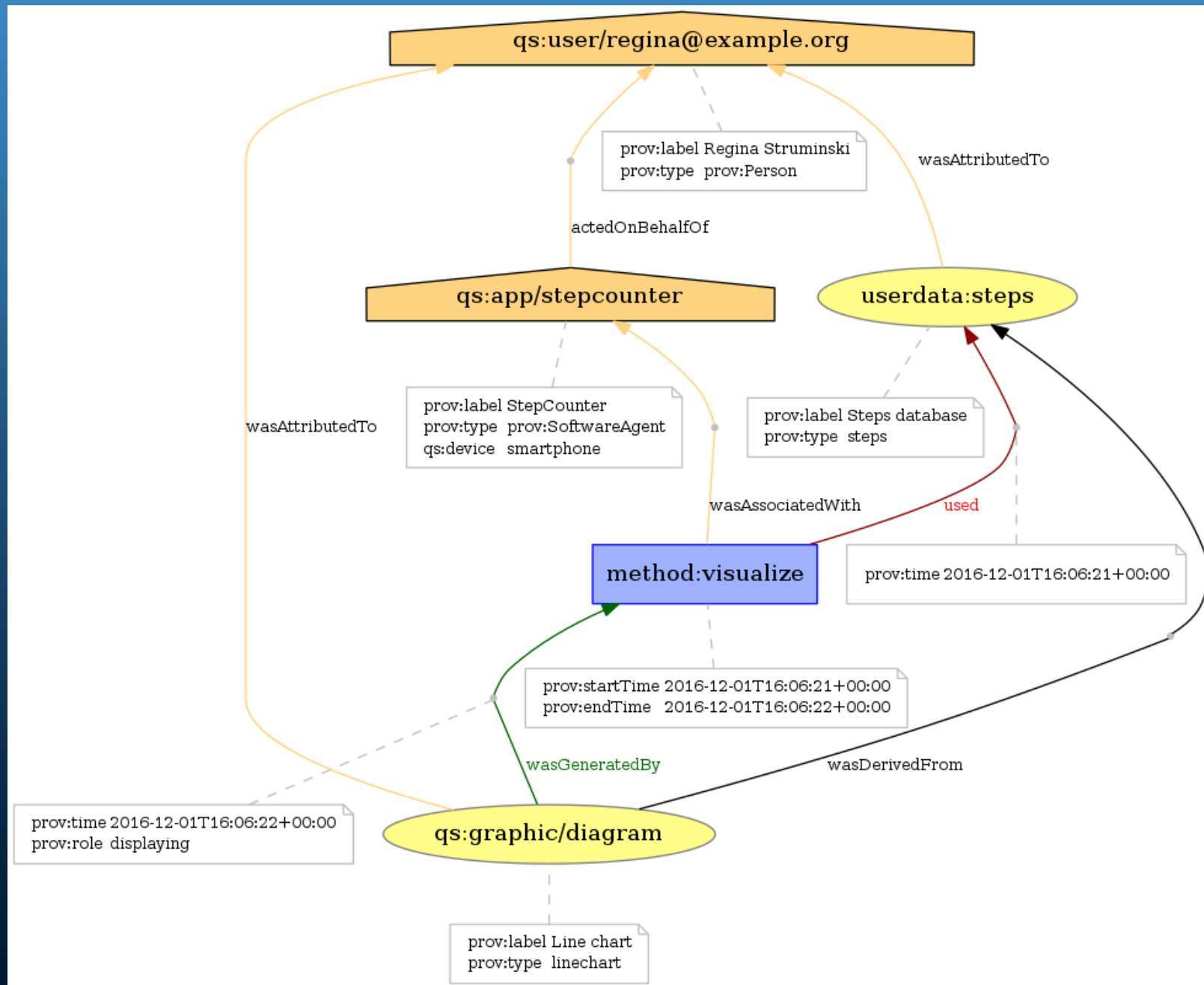
- Users
- Software
- Organizations

- Schreiber, A. (2016) ***A Provenance Model for Quantified Self Data***. In: Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices: 10th International Conference, UAHCI 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings, Part I, Springer, 382-393
- Schreiber A., Seider D. (2016) ***Towards Provenance Capturing of Quantified Self Data***. In: Provenance and Annotation of Data and Processes. IPAW 2016. Lecture Notes in Computer Science, vol 9672. Springer, Cham









document

```
prefix userdata <http://software.dlr.de/qs/userdata/>
prefix qs <http://software.dlr.de/qs/>
prefix graphic <http://software.dlr.de/qs/graphic/>
prefix app <http://software.dlr.de/qs/app/>
prefix user <http://software.dlr.de/qs/user/>
prefix device <http://software.dlr.de/qs/device/>
prefix method <http://www.java.com>
```

```
wasGeneratedBy(qs:graphic/diagram, method:visualize, 2016-12-01T16:06:22+00:00, [prov:role="displaying"])
activity(method:visualize, 2016-12-01T16:06:21+00:00, 2016-12-01T16:06:22+00:00)
entity(qs:graphic/diagram, [prov:type="linechart", prov:label="Line chart"])
entity(userdata:steps, [prov:type="steps", prov:label="Steps database"])
agent(qs:user/regina@example.org, [prov:type="prov:Person", prov:label="Regina Struminski"])
agent(qs:app/stepcounter, [prov:type="prov:SoftwareAgent", qs:device="smartphone", prov:label="StepCounter"])
wasAttributedTo(qs:graphic/diagram, qs:user/regina@example.org)
wasAttributedTo(userdata:steps, qs:user/regina@example.org)
actedOnBehalfOf(qs:app/stepcounter, qs:user/regina@example.org, -)
used(method:visualize, userdata:steps, 2016-12-01T16:06:21+00:00)
wasDerivedFrom(qs:graphic/diagram, userdata:steps, -, -, -)
wasAssociatedWith(method:visualize, qs:app/stepcounter, -)
```

endDocument

Standard Graph Visualizations and Textual Representations of Provenance Data are not Easy to Understand by Non-experts



Idea: Provenance Visualization Using Comics

Provenance Comics

- Presenting the provenance of processes in visual representation that people can understand without prior instructions or training (“Provenance for people”)
- Assumption
 - People are familiar with comics from every day life
 - See *daily strips* in newspapers etc.



Provenance Comics

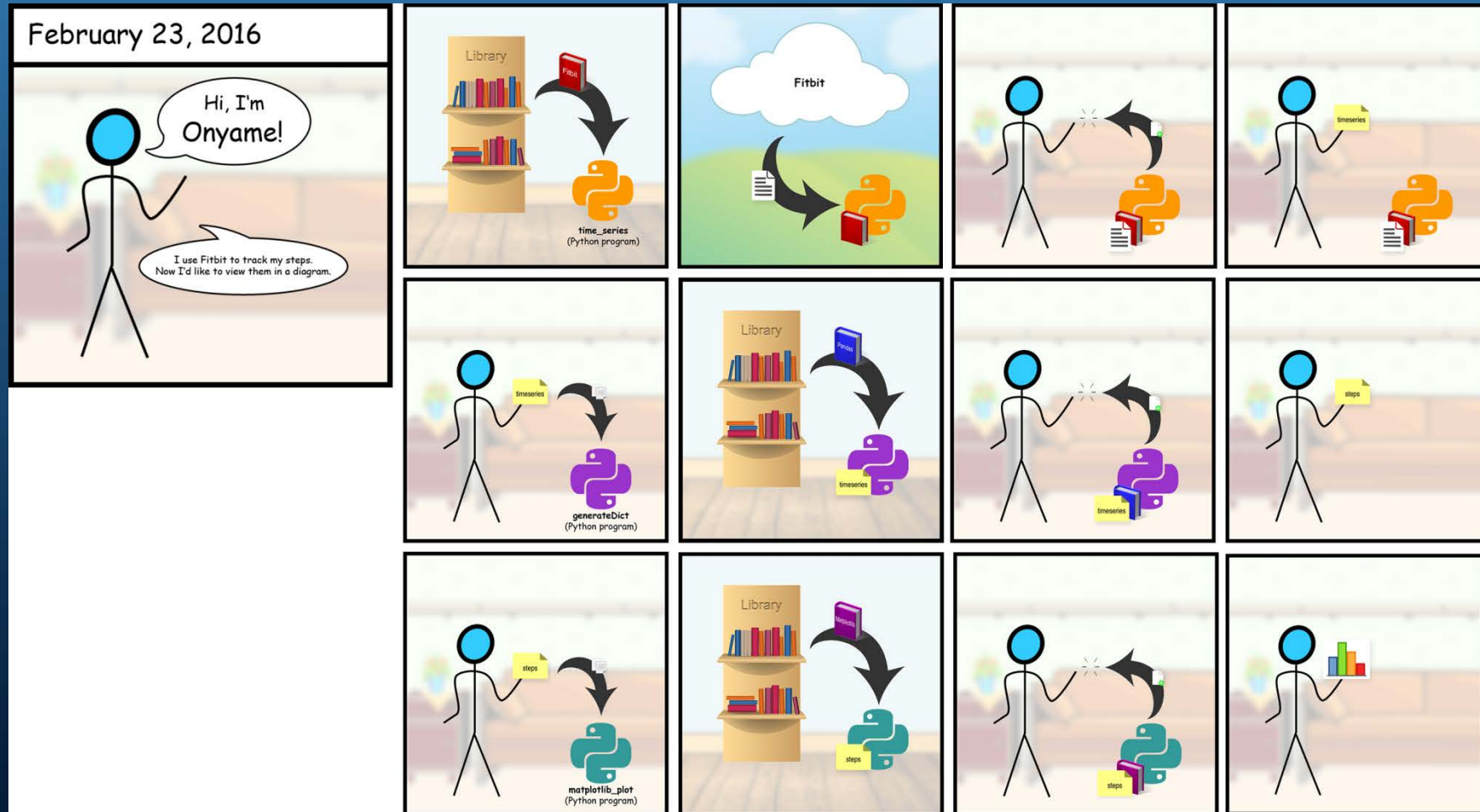
Design considerations

- Data provenance has a temporal aspect: origin, manipulation, transformation, and other activities happen sequentially over time
- The directed acyclic provenance graph guarantees that, while moving through its nodes, one always moves linearly forward in time
- It's possible to derive a temporal sequence of happenings from the graph that can be narrated like a story

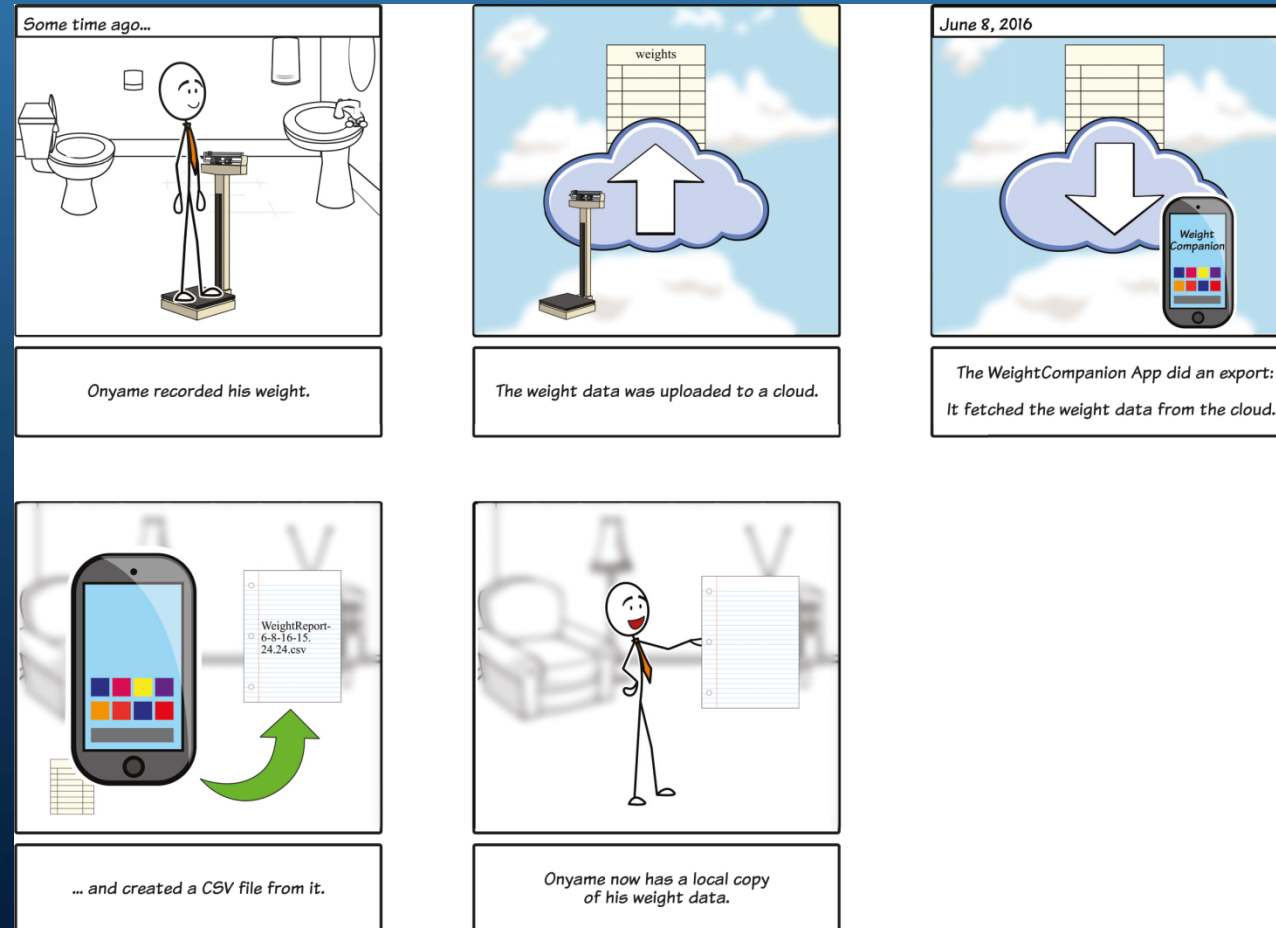
Mapping provenance graph to comics

- We generate a comic strip for each basic activity in the provenance graph
- Each strip consists of a varying number of panels, which are small drawings that provide further details about the activity
- The complete set of comic strips shows the “story” of the data

First Sketches



First Sketches



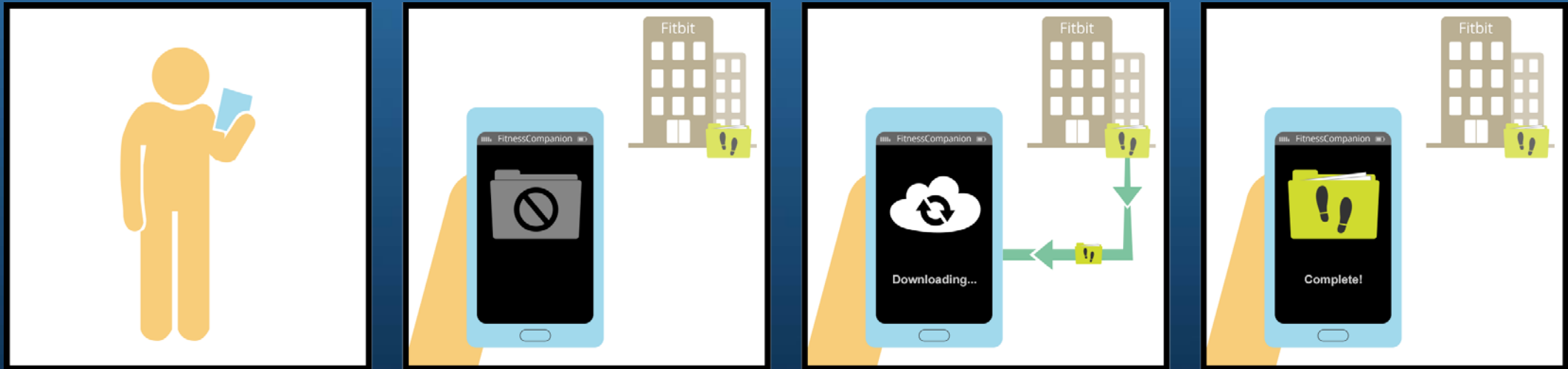
Current Graphical Style



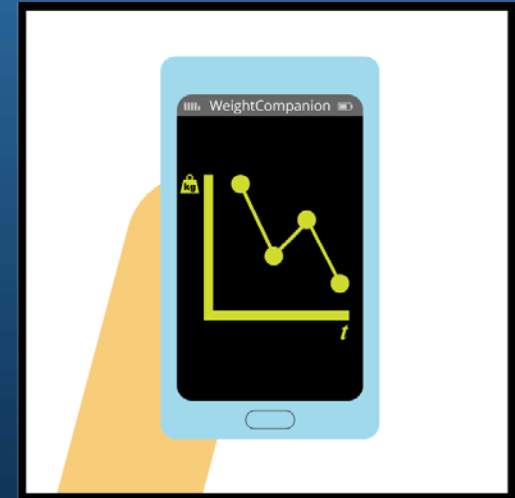
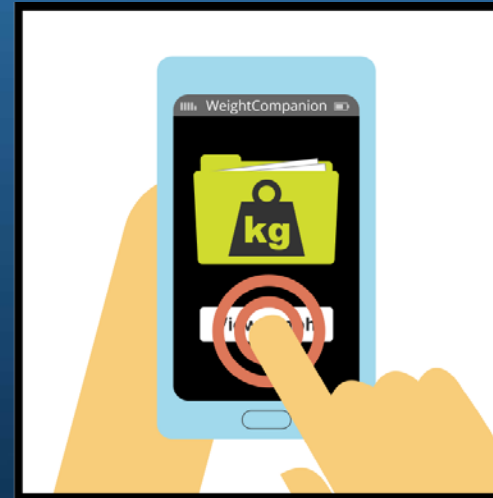
Single Comic Strip Shows a Single Data-related Action



Communicate to People Where Data is Stored

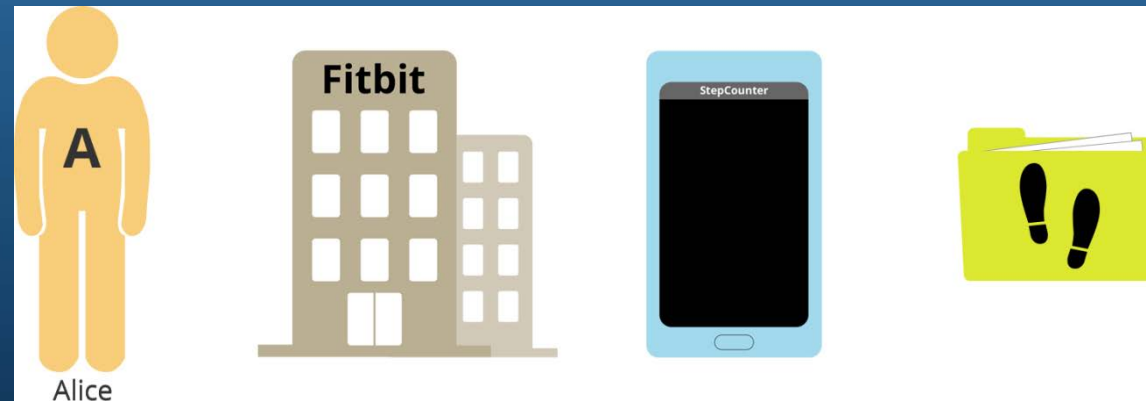


Understand How Data is Analyzed



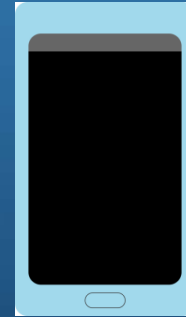
Distinctive Features

- Shapes
- Colors
- Icons
- Letters
- Labels



Representation of PROV Elements

Agents



Entities

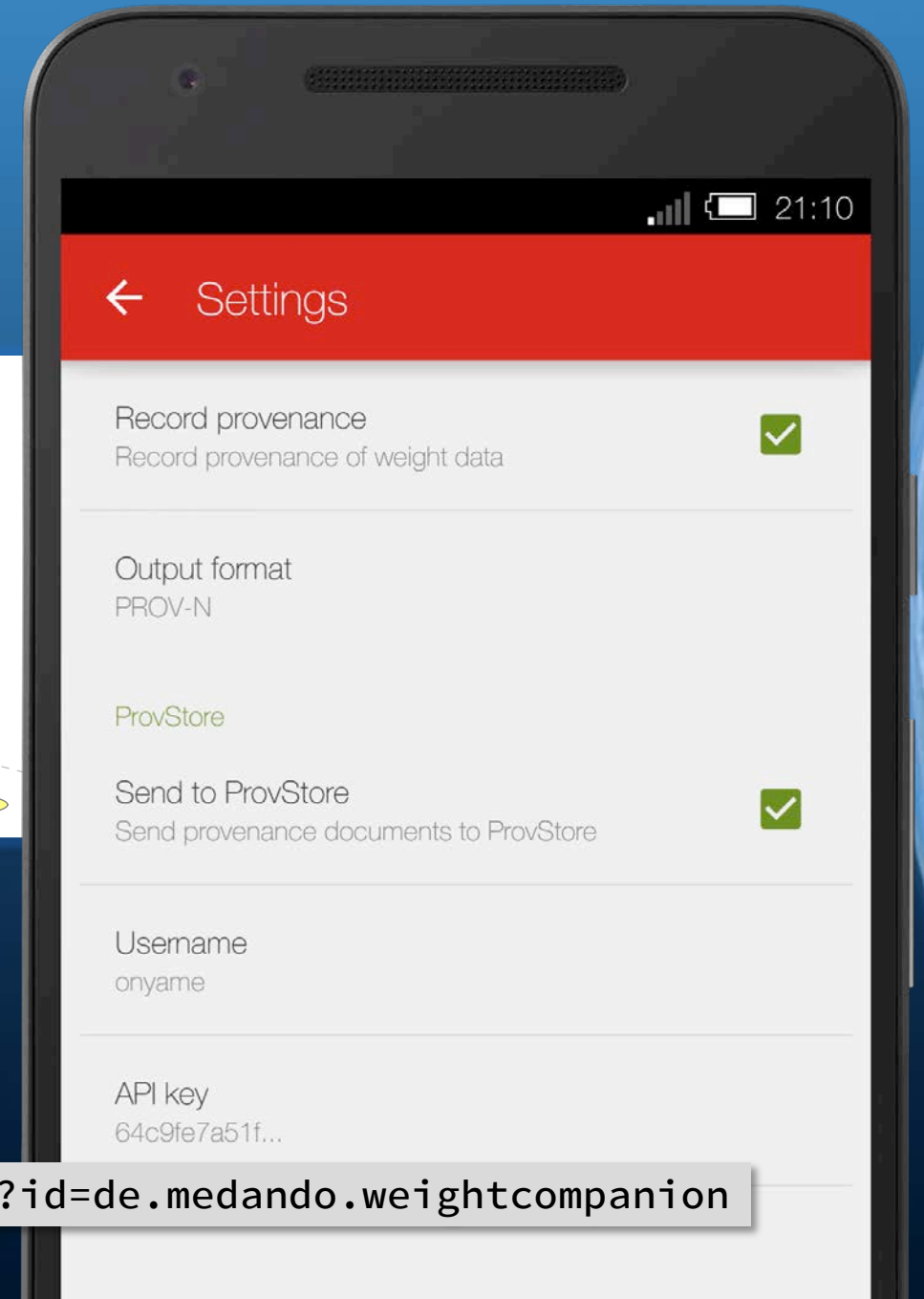
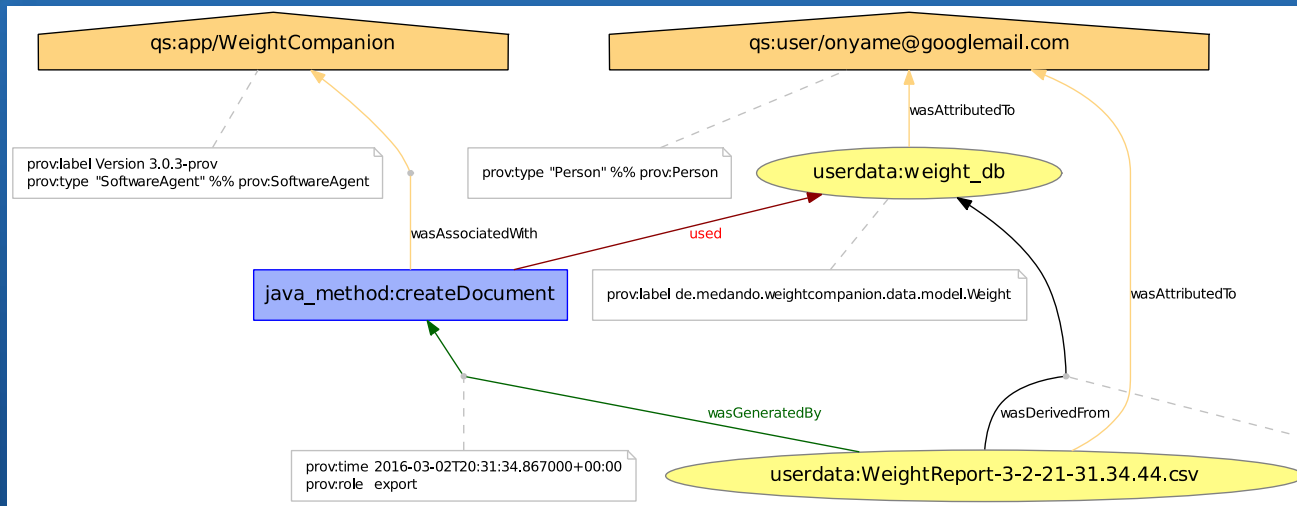


Activity-related

View graph

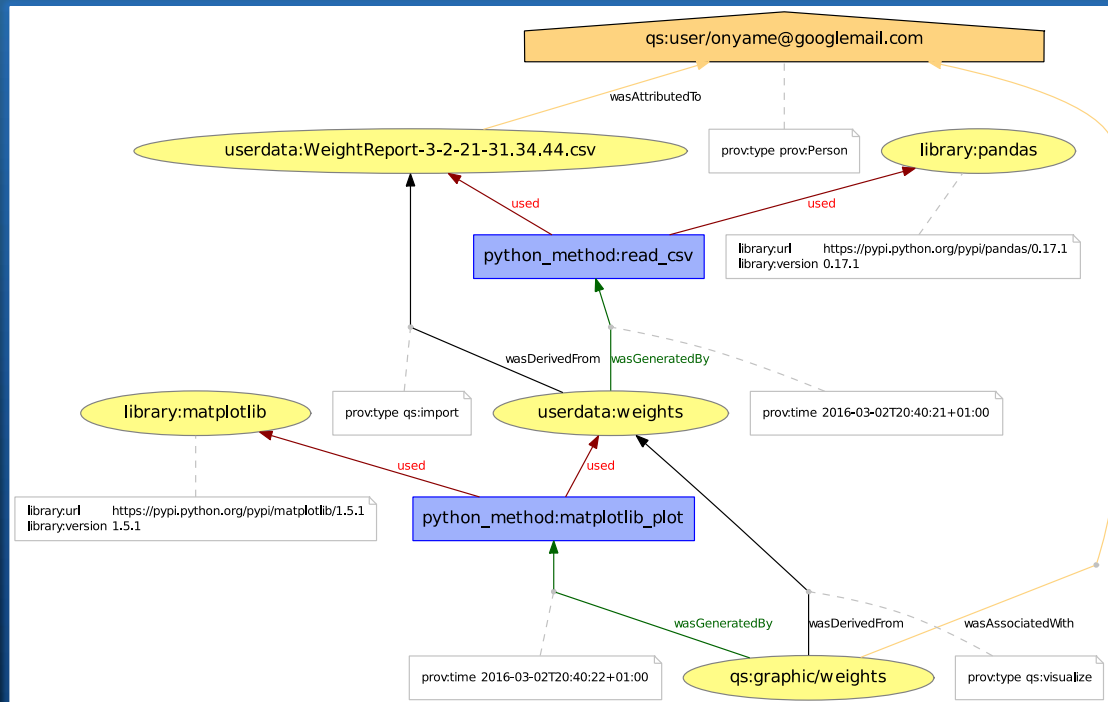


Collecting QS Provenance *Weight Tracking App*



<https://play.google.com/store/apps/details?id=de.medando.weightcompanion>

Collecting QS Provenance Visualization with Python Script



```
# Provenance-related Imports
from prov.model import ProvDocument, PROV
from provstore.api import Api
from time import gmtime, strftime

# Create a new provenance document
prov = ProvDocument()

# Add namespaces
prov.add_namespace('qs', 'http://software.dlr.de/qs/')
prov.add_namespace('userdata', 'http://software.dlr.de/qs/userdata/')
prov.add_namespace('user', 'http://software.dlr.de/qs/user/')
prov.add_namespace('graphic', 'http://software.dlr.de/qs/graphic/')
prov.add_namespace('library', 'https://pypi.python.org/pypi/')
prov.add_namespace('python_method', 'http://www.python.org')

# The user
agent_user = prov.agent('user:onyame@gmail.com', {'prov:type': PROV['Person']})

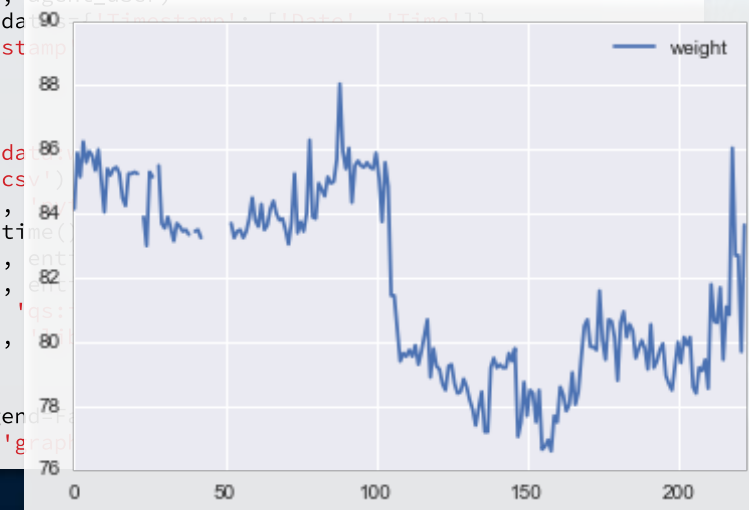
# Application Import
from pandas import DataFrame, Series, read_csv
import matplotlib.pyplot as plt

prov.entity('library:pandas', {'library:version': pd.__version__})
prov.entity('library:matplotlib', {'library:version': matplotlib.__version__})

# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:%s' % WC_FILE)
prov.wasAttributedTo(entity_csvfile, agent_user)
wc_data = read_csv(WC_FILE, parse_dates=[1],
                   index_col='Timestamp')

# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userdata:weights')
prov.activity('python_method:read_csv')
prov.wasGeneratedBy(entity_weights,
                    strftime('%Y%m%dT%H%M%S%Z', gmtime()))
prov.used('python_method:read_csv',
          prov.wasDerivedFrom(entity_weights,
                              other_attributes={'prov:type': 'qs:graphic/weights'}))
prov.used('python_method:read_csv',

# Plot the weights
weights.plot(title = 'Weight', legend=False)
entity_plot_weights = prov.entity('graphic:weights')
```



```
Date,Time,Weight,Waist,Hip,Device,Comment
"Jun 13, 2012",14:00,83.7,,Withings,
"Jun 13, 2012",14:08,79.7,,Withings,
"Jun 15, 2012",21:59,82.7,,Withings,
"Jun 15, 2012",22:04,82.7,,Withings,
"Jun 24, 2012",18:32,86.1,,Withings,
"Jun 26, 2012",07:42,80.8,,Withings,
"Jun 27, 2012",07:40,81.1,,Withings,
"Jun 29, 2012",07:34,79.4,,Withings,
"Jun 30, 2012",22:12,81.7,,Withings,
"Jul 1, 2012",11:21,80.6,,Withings,
"Jul 7, 2012",17:04,80.7,,Withings,
"Jul 10, 2012",07:46,81.8,,Withings,
"Jul 11, 2012",07:32,78.6,,Withings,
"Jul 12, 2012",07:26,79.4,,Withings,
```

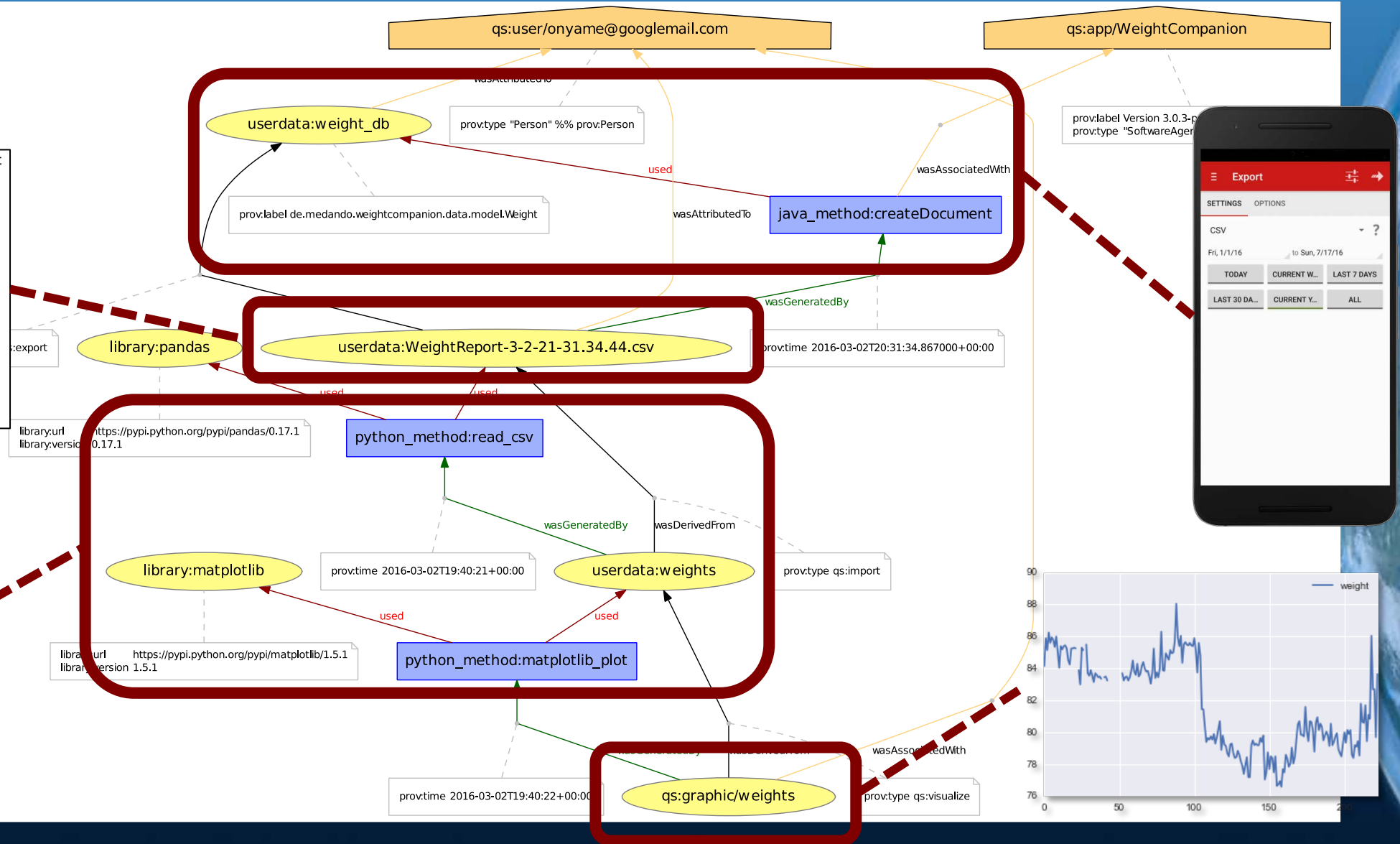
```
# Application Import
from pandas import DataFrame, Series, read_csv
import matplotlib.pyplot as plt

prov.entity('library:pandas', {'library:version': '0.17.1'})
prov.entity('library:matplotlib', {'library:version': '1.5.1'})

# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:weights', {'prov:label': WC_FILE})
prov.wasAttributedTo(entity_csvfile, agent_us)
wc_data = read_csv(WC_FILE, parse_dates={'Timestamp': [0, 1]}, index_col='Timestamp')

# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userdata:weights', {'prov:label': 'weights'})
prov.activity('python_method:read_csv', entity_weights)
prov.wasGeneratedBy(entity_weights, 'python_method:read_csv')
prov.strftime('%Y-%m-%dT%H:%M:%SZ', gmtime())
prov.used('python_method:read_csv', entity_weights)
prov.wasDerivedFrom(entity_weights, entity_csvfile, {'prov:type': 'qs:import'})
prov.used('python_method:read_csv', 'library:pandas')

# Plot the weights
weights.plot(title = 'Weight', legend=False)
entity_plot_weights = prov.entity('graphic:weights', {'prov:label': 'weights'})
prov.wasAssociatedWith(entity_weights, entity_plot_weights, {'prov:type': 'qs:visualize'})
```



On December 1, 2016 at 4:06 pm



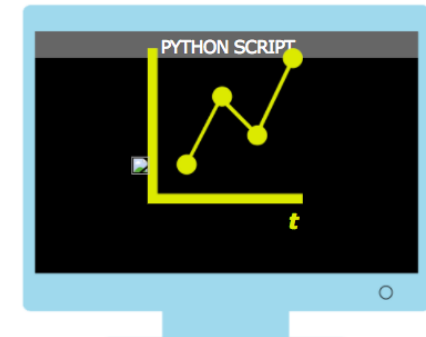
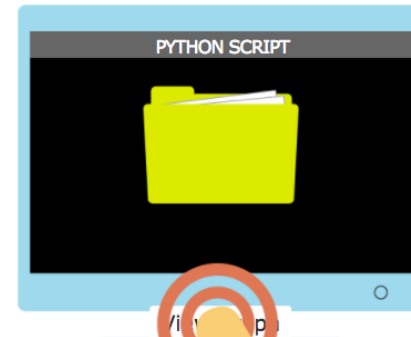
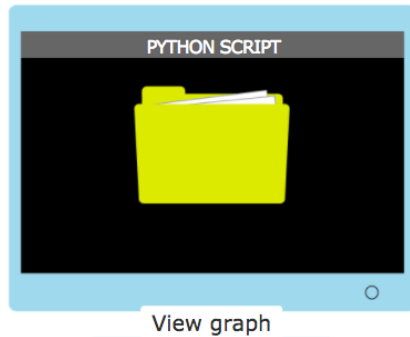
Andreas



On December 1, 2016 at 5:06 pm



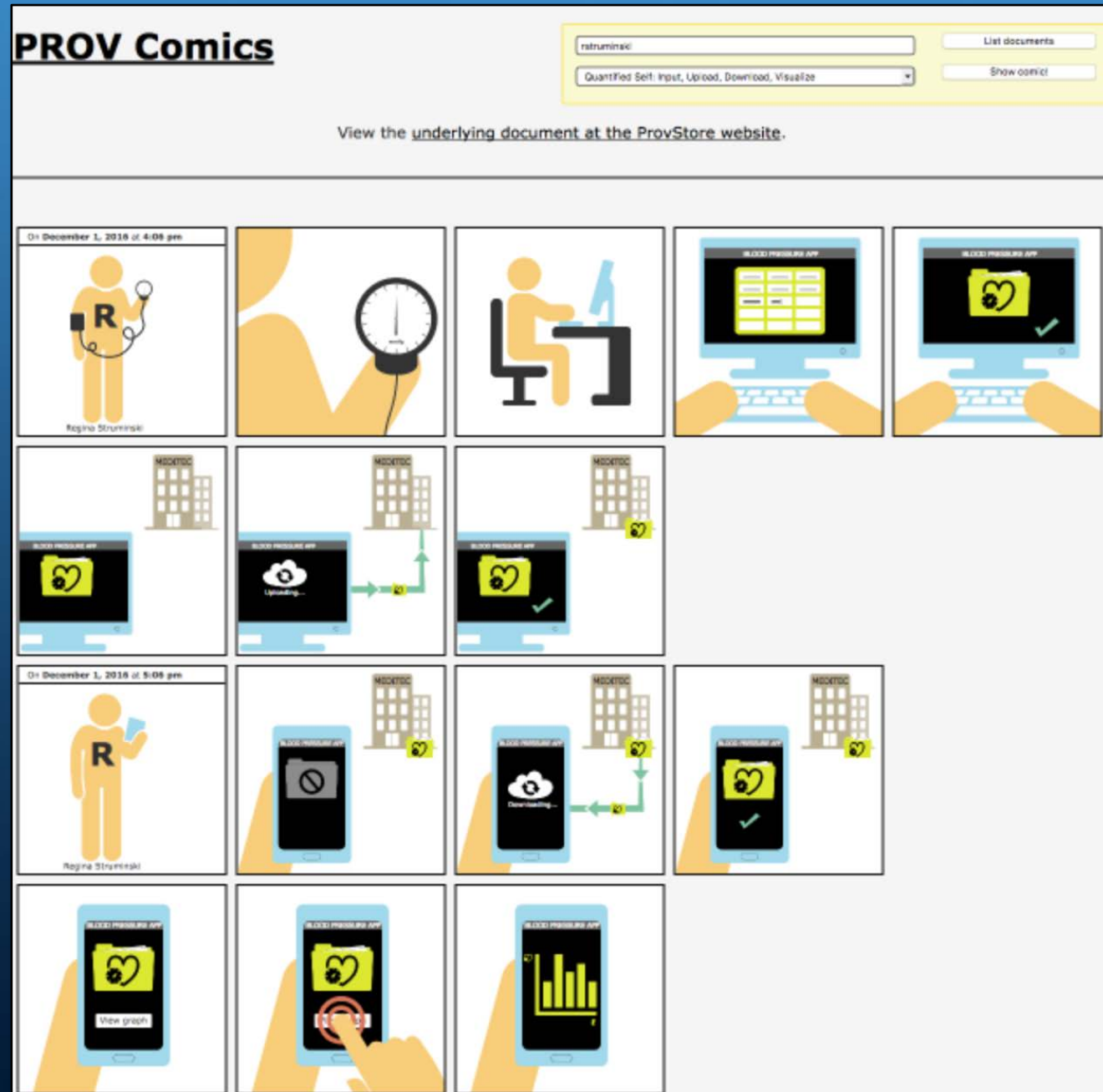
AndreasA



PROV Comics Web Application

<http://provcomics.de>

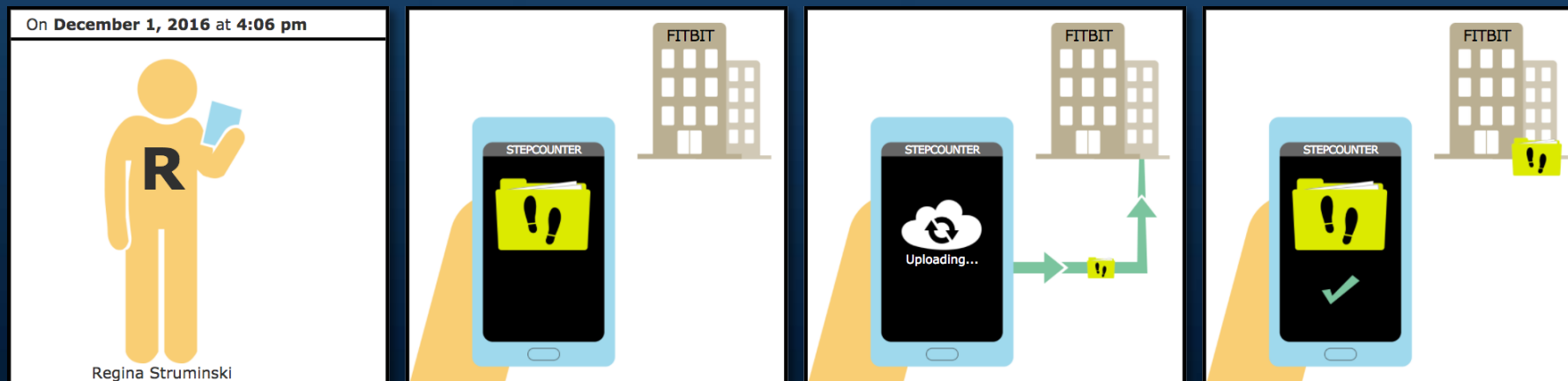
- Implemented in JavaScript
- Single page website
- Reads provenance graph from PROVSTORE
- Uses PROVSTORE jQuery API
- Code:
<http://github.com/DLR-SC/prov-comics>



Implementation Details

Additional attributes

```
agent(qs:app/stepcounter, [prov:type="prov:SoftwareAgent",
    qs:device="smartphone", prov:label="StepCounter"]])
agent(qs:service/fitbit,
    [prov:type="prov:Organization", prov:label="Fitbit"]])
wasGeneratedBy(userdata:activities/steps, method:request,
    2016-12-01T16:06:22+00:00, [prov:role="uploading"])
```



<http://provcomics.de/?username=rstruminski&docId=115547>

Open Issues

Current implementation is a prototype *with limitations*

- Flexibility and generalization
- Handling of
 - large provenance graphs
 - incomplete provenance data
 - branches and multiple data sources
- Expects a single PROV document

Future Work and Use Cases

Future Work

- Different comic styles
 - Comparative user studies
- Quantitative comics
 - Geographical information
 - Glyph-based depiction
- Technical improvements
 - Large Provenance graphs
 - Provenance templates
 - “Intelligent” generation of pictures

Possible Use Cases

- Journalism
- Generation of handbooks
- Communicating incidents

Thank You!

Andreas Schreiber

www.DLR.de/sc/ivs

andreas.schreiber@dlr.de

 [@onyame](https://twitter.com/onyame)